

# **Absolvování individuální odborné praxe**

## **Individual Professional Practice in the Company**

Václav Suchánek

Bakalářská práce

Vedoucí práce: RNDr. Eliška Ochodková, Ph.D.

Ostrava, 2021

## **Abstrakt**

Cílem této bakalářské práce je popsat mou práci, přínos a zkušenosti získané během mého působení v společnosti Y Soft. Během mé praxe jsem se věnoval testování aplikace YSoft SAFEQ 6 pro hardwarový terminál Terminal Pro 4 zaměřené na tiskové úlohy. Náplní mé práce bylo navrhování, implementace a úprava testů pro robotickou ruku za pomoci interního systému RMS, Robot Frameworku s použitím Gherkin syntaxe a jazyku Python. Součástí mé práce bylo také ověřování stability testů a případné úpravy ke zvýšení stability, optimalizace rozpoznávání obrazovek za pomoci umělé inteligence a definice elementů na obrazovce.

**Klíčová slova:** automatizace, testování, odborná praxe, Python, Robot Framework, Gherkin, Y Soft, bakalářská práce

## **Abstract**

The goal of this bachelor thesis is to describe my work, benefit and experiences gained during my practice in Y Soft corporation. During my practice, I focused on testing YSoft SAFEQ 6 application for hardware terminal Terminal Pro 4. My job consisted of designing, implementing, and modifying tests for robotic arm using internal RMS system, Robot Framework with Gherkin syntax, and programming language Python. I also had to verify test stability, modify said tests to increase stability if needed, optimize screen recognition using artificial intelligence and define elements contained on screens.

**Key words:** automation, testing, individual practice, Python, Robot Framework, Gherkin, Y Soft, bachelor thesis

Rád bych poděkoval mé vedoucí RNDr. Elišce Ochodkové, Ph.D. za rady a vedení při tvorbě bakalářské práce, mému konzultantovi Bc. Martinu Kubalovi za nápomocnost a trpělivost, mým kolegům za poskytování cenných informací a celé společnosti Y Soft za příležitost.

## Obsah

Seznam zkratk .....	5
Seznam obrázků .....	6
Seznam zdrojových kódů .....	7
1. Úvod.....	8
2. Y Soft .....	9
2.1 O Společnosti .....	9
2.2 O Týmu .....	9
3. Úkoly zadané během praxe .....	10
4. Použitý SW a HW .....	11
4.1 YSoft SAFEQ 6 .....	11
4.2 RMS .....	11
4.3 IMPROC .....	13
4.4 Terminal Pro 4.....	13
4.5 Python .....	14
4.6 PyCharm .....	14
4.7 Robot framework.....	14
4.8 JIRA.....	14
4.9 Sourcetree .....	14
4.10 Bamboo .....	14
5. Vypracování úkolů zadaných během praxe.....	15
5.1 Příprava .....	15
5.2 Autentizace .....	18
5.2.1 Jednofaktorová.....	22
5.2.2 Dvoufaktorová.....	25
5.2.3 Více možností autentizace .....	28
5.2.4 Přiřazení čipové karty .....	29
5.3 Tisk .....	30
5.4 Kopírování.....	34
5.5 Skenování .....	35

5.6	Nastavení terminálu.....	36
6.	Závěr .....	38
7.	Literatura.....	39

## Seznam zkratek

CI/CD	- Continuous Integration/Continuous Deployment
DHCP	- Dynamic Host Configuration Protocol
HTML	- HyperText Markup Language
JSON	- JavaScript Object Notation
LCD	- Liquid Crystal Display
PIN	- Personal Identification Number
RMS	- Robotic management system
SQ	- SAFEQ
XML	- Extensible Markup Language

## Seznam obrázků

1	Dialogové okno semiautomatického testu	15
2	Elementy na obrazovce „Login screen“, varianta „PIN“	21
3	Flow autentizace pomocí pinu pro Terminal Pro 4	22
4	Elementy na obrazovce klávesnice	23
5	Flow přihlášení pomocí uživatelského jména a hesla	25
6	Elementy na obrazovce tisku bez zvolené tiskové úlohy	28
7	Elementy na obrazovce detail úlohy	30
8	Obrazovka na Terminal Pro 4 při kopírování	31
9	Obrazovka volby skenovacího vzoru	32
10	Obrazovka probíhajícího skenování	33

## **Seznam zdrojových kódů**

1	Příklad testu v Robot Frameworku s Gherkin syntaxí	17
2	Příklad *.robot souboru	20



# 1. Úvod

Cílem mé bakalářské práce bylo vykonání odborné praxe ve společnosti Y Soft, která se zaměřuje na tisková řešení pro střední a velké podniky. Vykonání bakalářské práce formou odborné praxe jsem si vybral pro získání zkušeností s prací v týmu, nahlédnutí do vývojových procesů větší firmy, získání znalostí nad rámec těch, vyučovaných na VŠB-TUO a ověření v praxi těch, které jsem během mého studia na VŠB-TUO již získal. Pro vykonání mé praxe právě ve společnosti Y Soft jsem se rozhodl na základě renomé společnosti a skutečnosti, že byla založena jako studentský projekt. Téma odborné praxe jsem si vybral na základě mé osobní fascinace roboty a umělou inteligencí.

Do společnosti jsem byl přijat na základě pracovního pohovoru, během kterého byly ověřeny a posouzeny mé znalosti, zkušenosti a má schopnost vykonávat případnou zadanou práci. Po přijetí jsem se stal členem jednoho ze dvou týmů ostravské pobočky společnosti Y Soft.

Popis mé práce je rozdělen do pěti kapitol. První z nich je věnována představení společnosti Y Soft a fungování týmu, jehož členem jsem byl. Ve druhé kapitole jsou představeny úkoly, které mi byly zadány v průběhu mé praxe, ve třetí pak specializovaný software a hardware, který jsem k během své odborné praxe využíval. Čtvrtá kapitola je věnována popisu vypracování zadaných úkolů, poslední kapitola shrnuje mou práci a zkušenosti při ní získané či chybějící.

## 2. Y Soft

### 2.1 O Společnosti

Společnost Y Soft [\[1\]](#) byla založena v roce 2000 v rámci studentského projektu na Masarykově univerzitě v Brně. V současné době má společnost více než 300 zaměstnanců, zastoupení v 16 zemích a více než 14 000 zákazníků z více než 120 různých zemí. Produkty Y Soft jsou používány také řadou společností, patřících do žebříčku Fortune 500.

Jejím hlavním zaměřením je vývoj tiskových řešení, jejichž cílem je zvýšení efektivity, bezpečnosti tisku a snížení tiskových nákladů. Společnost vlastní také výrobní oddělení, pomocí kterého je schopna výroby vlastních zařízení a dílů.

### 2.2 O Týmu

Během mé praxe ve společnosti Y Soft jsem patřil do jednoho z ostravských týmů, nazývaný Sova. Tento tým se skládá ze 7 stálých členů a je jedním z řady týmů majících na starost vývoj YSoft SAFEQ 6.

Organizace týmu je založená na metodice LESS Scrum, což je agilní metodika řízení vývoje softwaru. Práce je rozdělena do čtrnáctidenních úseků zvaných sprinty, na jejichž konci by měl tým dokončit předem určené množství práce. Každé ráno se členové týmu setkávají a navzájem se informují o stavu přiřazené práce, v případě potřeby si navzájem poradí či pomůžou. Na konci každého sprintu se tým setká, zhodnotí svou práci a přiřadí si novou pro další sprint.

### 3. Úkoly zadané během praxe

Cílem mé práce ve společnosti Y Soft bylo:

- **Automatizace testů pro Terminal Pro 4**

Mým hlavním úkolem bylo vytvoření, implementace a úprava sady testů pro automatizované testování Terminal Pro 4 za pomoci robota, vyvíjeného společností Y Soft. Má práce byla velmi odlišná od práce mých předchůdců zejména proto, že se jedná o externí hardwarový terminál schopný s tiskárnou pouze komunikovat, ne ji přímo ovládat. Bylo proto zapotřebí již existující testy a knihovny upravit tak, aby bylo i přes tuto skutečnost možné provádět automatizované testování na Terminal Pro 4. Hlavními testovanými oblastmi byly:

- Autentizace
- Autentizovaný a přímý tisk
- Kopírování
- Skenování
- Zaúčtování a vyúčtování
- Nastavení terminálu

- **Vytvoření možnosti spolupráce dvou robotů**

Jak bylo již řečeno v předchozím bodě, Terminal Pro 4 není schopen tiskárnu přímo ovládat. Pro testování některých funkcí je tedy zapotřebí druhého robota umístěného u tiskárny, se kterou Terminal Pro 4 komunikuje. Tito dva roboti poté spolupracují k dosažení požadovaného výsledku. Vzhledem k prvenství tohoto požadavku bylo tedy jedním z mých úkolů vytvoření možnosti této spolupráce.

- **Ověření stability testů**

Testování pomocí robota je z podstaty mnohem více náchylné k chybám než softwarové testování. Na rozdíl od něj totiž může na robota působit řada vnějších vlivů, které mohou výsledek testu negativně ovlivnit. Mezi nejčastější příčiny falešně negativních testů patří již zmíněná komunikace mezi Terminal Pro 4 a tiskárnou, nesprávná kalibrace robota či kamery, problémy tiskárny vyřešitelné jednoduše restartem a chyby způsobené uživatelem. Mým cílem bylo tedy mimo jiné zvýšit stabilitu testů, bylo-li to možné, případně vylepšit předání informací o neúspěchu uživateli tak, aby byl důvod neúspěchu testu co nejvíce pochopitelný.

- **Ušetření času**

Hlavním cílem automatizace testů je ušetřit čas software inženýrům. Na rozdíl od nich se totiž robotická ruka neunaví a může pracovat samostatně, bez dozoru i mimo běžné pracovní hodiny. Software inženýři tak můžou čas, který by strávili testováním, věnovat dalšímu vývoji. Testování stále stejných scénářů je pro člověka únavné, zato pro robota ideální.

## 4. Použitý SW a HW

### 4.1 YSoft SAFEQ 6

YSoft SAFEQ 6 [\[2\]](#) (dále jen SAFEQ) je řešení sloužící k řízení, optimalizaci a zabezpečení tiskových a digitálních procesů vyvíjené společností Y Soft. SAFEQ je mimo jiné využíváno také na VŠB-TUO. Mezi jeho hlavní výhody patří:

- **Statistiky tisku**  
SAFEQ umožňuje sledovat a analyzovat využití zařízení podle uživatelů, oddělení nebo umístění a provádět audity bezpečnostních rizik a trendů využití zařízení.
- **Tisk metodou pull**  
Tiskové úlohy nejsou posílány přímo do zařízení, ale na server, díky čemuž mohou být vytištěny z libovolného zařízení a bez nutnosti instalace ovladačů.
- **Bezpečnost dokumentů**  
Nutnost přihlášení před přístupem k funkcím tiskárny snižuje riziko zanechání potencionálně citlivých dokumentů v tiskárně a nahlédnutí do dokumentů před příchodem k tiskárně.
- **Scan Workflows**  
Jedná se o vzory definované v jazyce XML, které umožňují velice uživatelsky přívětivé skenování dokumentů. Jejich použitím docílíme toho, že po jediném kliknutí je například zaslán naskenovaný soubor na e-mail uživatele nebo uložen na disk.
- **Účtování**  
SAFEQ podporuje placené služby a kvóty stránek tak, aby měl vlastník tiskárny plnou kontrolu nad výdaji.
- **Pravidla tisku**  
SAFEQ umožňuje zavést pravidla pro tiskové úlohy, což umožňuje jednoduše uživatelům či skupině uživatelů zakázat například barevný či oboustranný tisk a dává tak majiteli zařízení možnost nadále snížit náklady na provoz.

### 4.2 RMS

RMS je interní systém, který je neodmyslitelnou součástí robotického systému. Pomocí jeho webového rozhraní může uživatel spravovat roboty, kalibrovat je a přidávat a upravovat údaje, které robot ke své funkci potřebuje. Mezi nejvýznamnější zmíněné údaje patří:

- **Screens**  
Fotky obrazovky zařízení v určitém momentu, které jsou později použity k nezbytné identifikaci obrazovky, na které se zařízení právě nachází. Pod jedním referenčním jménem

může být uloženo více obrazovek. Jsou-li si dvě či více obrazovek vzhledově podobné, lze použít tzv. distinguishers, které za pomoci rozlišení textu či vzhledové podobnosti oblasti rozliší, o kterou obrazovku se jedná. Díky této funkci lze při rozlišování obrazovek dosáhnout téměř stoprocentní přesnosti. Každá obrazovka má příznak, jestli je k jejímu dosažení potřeba provést autentizaci či nikoliv, a stav určující požadované chování robota, narazí-li na tuto obrazovku.

- **Elements**

Elementy jsou aktivní prvky, definované zvlášť pro každou obrazovku. Dělíme je na hardwarové čili fyzická tlačítka mimo obrazovku a softwarové, vykreslené na obrazovce. U každého elementu lze definovat několik vlastností:

- **Transitions**

Určuje, na jakou obrazovku by se robot po stisknutí tlačítka měl dostat. Cílových obrazovek může být více. Transitions jsou užitečné, má-li robot ověřit, zda skutečně ke stisknutí tlačítka došlo a dostal se po jeho stisknutí na správnou obrazovku. Další významnou úlohou transitions je automatická navigace robota na zadanou obrazovku.

- **Positioning**

Tato vlastnost informuje robota, jestli je element statický, tedy vždy na stejném místě určeném uživatelem, nebo je dynamický a jeho poloha se mění. V případě dynamické polohy je třeba zadat text, pomocí kterého je tlačítko nalezeno.

- **Grouping**

Grouping se používá pro skupiny elementů se stejnou funkcí a je užitečné, potřebuje-li robot znát stav dalších elementů ze skupiny. Nejčastější využití této vlastnosti je pro seznamy položek, ikony v menu nebo složky a záložky.

- **States**

Tato vlastnost je použita u elementů, které lze nějakým způsobem označit – je třeba přidat obrazovku, kde je element aktivní a obrazovku, kde aktivní není. Tato vlastnost je velmi důležitá pro stabilitu testů, protože robotovi umožňuje detekovat stav elementu, případně jej změnit a změnu ověřit.

- **Special functions**

Poslední vlastnost obsahuje několik možností elementu, ovlivňující především navigaci:

- **Paging next, Paging previous**

Jedná se o vzájemně exkluzivní možnosti používané u obrazovek, jejichž obsah je příliš velký a je rozdělen do několika stránek. Označením elementu těmito

možnostmi robotovi oznamujeme, že po kliknutí na element se dostane na další či předchozí stránku.

- **Vertical scrolling, horizontal scrolling**

Opět vzájemně exkluzivní možnosti, oznamující robotovi skutečnost, že se dá v elementu pohybovat horizontálním či vertikálním táhnutím.

- **Ignore in path finding**

Při označení této možnosti bude robot element ignorovat při automatické navigaci na obrazovku a klikne na něj tedy pouze bude-li mu to výslovně přikázáno. Tato možnost je důležitá pro stabilitu testů, protože chování některých elementů může být nepředvídatelné nebo se měnit v závislosti na nastavení.

- **Flows**

Flow označuje sled akcí, jako například emulace karty, a kliknutí na elementy specifických pro danou tiskárnu a aplikaci (viz obrázek 3 a 5). Použití flows řeší problém různých postupů pro dosažení stejných výsledků na různých tiskárnách a umožňuje tak používat stejné testy pro různá zařízení. Každá flow má příznak, je-li autentizační. V případě, že je nastavena jako autentizační, ověří systém po dokončení flow, zda se nachází na obrazovce, pro jejíž dosažení je potřeba autentizace. Pro zvýšení stability testování může mít každý krok flow nastaveno ověření obrazovky. Po dokončení stisknutí elementu tak robot ověří, zda se dostal na obrazovku definovanou ve vlastnosti transitions. V případě, že ověření obrazovky selže, robot zopakuje předchozí krok flow a znovu obrazovku ověří.

## 4.3 IMPROC

IMPROC je další nedílnou součástí interního robotického systému. Díky této části systému je robot schopen rozeznávat obrazovky, texty, barvy a další potřebné vizuální elementy. IMPROC ke své funkci využívá umělou inteligenci typu umělá neuronová síť. Vzhledem k velké náročnosti systému na výpočetní výkon je IMPROC spuštěn na samostatném počítači, s kamerou a dalšími potřebnými částmi systému komunikuje pomocí síťového připojení.

## 4.4 Terminal Pro 4

Terminal Pro 4 je hardwarový terminál vyvinutý společností Y Soft používaný zejména při potřebě použití systému SAFEQ na tiskárnách, u kterých není použití aplikace SAFEQ na zabudovaném terminálu tiskárny žádoucí či možné. Terminal Pro 4 může být s tiskárnou spojen tzv. blokovacím kabelem, který zamezí použití tiskárny jinak než pomocí terminálu.

## 4.5 Python

Python [\[3\]](#) je open-source, interpretovaný, objektově orientovaný, vysokoúrovňový programovací jazyk s dynamickou sémantikou. Poprvé byl vydán v roce 1991, v posledních letech se těší velké popularitě a již několik let po sobě se řadí mezi tři nejpopulárnější programovací jazyky [\[4\]](#). Za svou oblíbenost vděčí hlavně jednoduché syntaxi a rozsáhlým zabudovaným knihovnám.

## 4.6 PyCharm

Pycharm [\[5\]](#) je vývojové prostředí pro Python, vyvíjené českou firmou JetBrains.

## 4.7 Robot framework

Robot framework [\[6\]](#) je open source framework pro automatizaci testů a robotických procesů. Vyznačuje se jednoduchou syntaxí s jednoduše pochopitelnými klíčovými slovy. Je implementovaný a rozšiřitelný pomocí Pythonu, Jythonu (Java) nebo IronPythonu (.NET).

## 4.8 JIRA

JIRA [\[7\]](#) je softwarový nástroj pro evidenci chyb a problémů při vývoji softwaru nebo řízení projektů vyvíjený společností Atlassian.

## 4.9 Sourcetree

Sourcetree [\[8\]](#) je git klient vyvíjený společností Atlassian. Jelikož se jednalo o mou první zkušenost s vývojem v týmu a použitím gitu, bylo mi mým vedoucím doporučeno použití tohoto klienta, který velmi urychlil mé chápání celého systému.

## 4.10 Bamboo

Bamboo [\[9\]](#) je klient pro CI/CD pipeline vyvíjený společností Atlassian. S jeho pomocí je možné zlepšit proces vývoje nových verzí aplikace, zejména ve fázi integrace, a automatizovat testování.

## 5. Vypracování úkolů zadaných během praxe

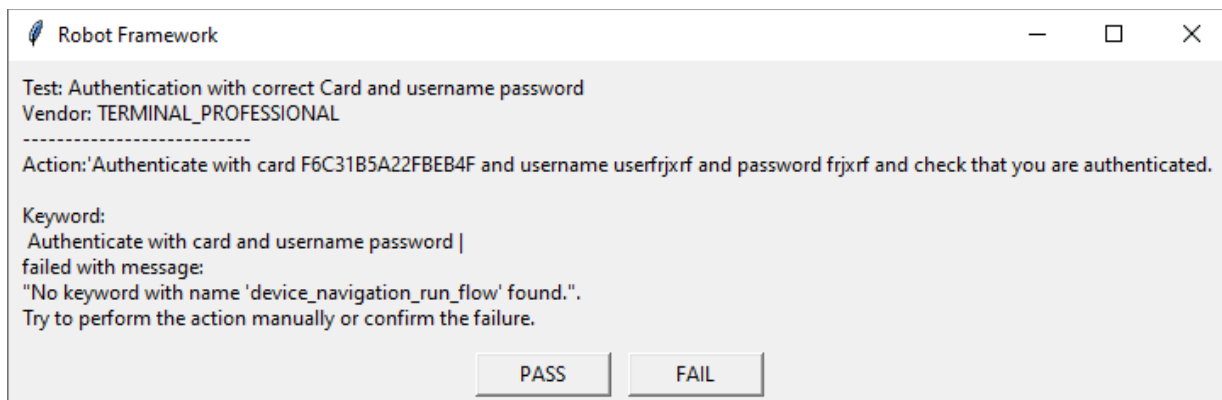
### 5.1 Příprava

Mým úplně prvním úkolem ve společnosti bylo seznámení se s pravidly bezpečnosti práce a požárními předpisy, které probíhalo za pomoci prezentace v interním systému a bylo zakončeno testem. Po úspěšném absolvování a podepsání certifikace jsem byl seznámen se společností Y Soft a systémem SAFEQ několika instruktážními videi, z nichž každé bylo věnováno specifickému tématu či problematice.

Dalším krokem byla instalace a připravení všech programů potřebných pro práci a seznámení se s nimi. Přestože jsem měl s většinou z nich zkušenosti, hlavně pak z mého studia na VŠB, se dvěma důležitými částmi mé práce jsem se doposud nesetkal. Jednalo se konkrétně o systém Git a framework Robot Framework používaný pro implementaci testů. S fungováním systému Git, konkrétně pak jeho grafickou nástavbou Sourcetree, jsem byl důkladně obeznámen mým konzultantem, informace o Robot Frameworku jsem čerpal zejména z rozsáhlé oficiální dokumentace frameworku, popřípadě již napsaných testů. Členové týmu, do kterého jsem patřil, ale také zaměstnanci převážně z brněnské pobočky společnosti Y Soft byli v této fázi velmi velkou podporou a zdrojem drahocenných informací. Poté jsem si vytvořil vlastní virtuální prostředí a nainstaloval na něj systém SAFEQ, aby nebyly výsledky testů ovlivňovány akcemi ostatních spolupracovníků.

Dalším krokem bylo seznámení s Terminal Pro 4 pomocí semiautomatických neboli poloautomatických testů. Jedná se o druh testů, které ke svému fungování nepotřebují robota, ale vyžadují lidskou obsluhu. Absence robota znamená, že do systému není potřeba přidávat obrazovky, elementy a flows, což z krátkodobého hlediska ušetří velké množství času. Princip těchto testů spočívá v tom, že jsou uživateli předávány pomocí dialogového okna přesné instrukce, které po vykonání potvrdí nebo zamítne, objeví-li se nežádoucí chování (viz obrázek 1). Člověk při nich tedy plně nahrazuje funkci robota, což je z dlouhodobého hlediska časově nevýhodné. Výhodou těchto testů je to, že jsou aplikovatelné na jakékoliv zařízení a jsou přesně skriptované, testerovi se tudíž nemůže stát, že na nějaký test zapomene nebo jej vykoná špatně, jak by se mohlo stát v případě, že by testy vykonával bez navádění Robot Frameworkem. Další výhodou oproti neasistovanému testování je to, že program sám vykoná všechny funkce, které by vykonal při plně automatických testech. Tester tak nemusí například před každým testem vytvářet v systému uživatele, přiřazovat mu přihlašovací údaje nebo třeba kontrolovat stav tiskové úlohy, což výrazně zkrátí dobu testování a do jisté míry z něj odstraní případné lidské chyby. Každý, tedy i plně automatický test, by měl být spustitelný v poloautomatickém režimu, ale ne každý poloautomatický test může být plně automatický.





Obrázek 1: Dialogové okno semiautomatického testu

Po seznámení se se všemi potřebnými systémy jsem byl připraven zahájit práci na zadaných úkolech. Před samotnou implementací a úpravou testů však bylo potřeba připravit potřebný hardware.

Robot se skládá z několika částí:

- **Robotická ruka**

Nezbytnou částí robota je samotná robotická ruka, která mu umožňuje simulovat interakci člověka a pomocí doteků komunikovat s terminálem a hardwarovými tlačítky. Ruka se skládá ze tří kloubů, díky kterým se může libovolně natáčet a dotknout se tak obrazovky kdekoli a pod jakýmkoliv úhlem. Na základně robotické ruky je rotační kloub, který umožňuje rotaci robotické ruky o 180° a díky kterému se ruka může dotknout obrazovky i v jejich rozích. Zakončená je stylusem, což je gumový prvek schopný simulovat elektrický náboj lidského těla, díky kterému jsou registrovány doteky na dnes téměř výhradně používaných kapacitních dotykových obrazovkách. Gumový povrch prvku pak zajišťuje, že nedojde k poškození či poškrábání obrazovky.

- **Kamera**

Dalším potřebný prvkem je kamera, pomocí které je zajištěno rozeznávání obrazovky. Kamera je vyvýšená nad robotickou ruku dřevěným stojanem vyráběným společností Y Soft, na kterém se nachází kamerový stativ se třemi klouby, umožňující natočit kameru v libovolném směru a úhlu. Díky tomuto stativu jsem mohl natočit kameru kolmo k obrazovce a vyhnout se tak případným chybám obrazu, pramenícím z převodu lichoběžníkového snímku obrazovky na obdélníkový. Všechny snímky obrazovky uvedené v této práci jsou vyfoceny pomocí této kamery a jsou tedy přesným zobrazením toho, jak vidí obrazovku robot.

- **Emulátor čipových karet**

Terminal Pro 4 podporuje přihlášení pomocí čipové karty, je-li k němu připojena čtečka karet, proto je robot vybaven emulátorem čipových karet, díky kterému můžeme

simulovat přiložení jakékoliv čipové karty. Při testech, které neověřují určitý způsob přihlášení, je přihlášení pomocí karty používáno jako výchozí způsob přihlášení, protože je nejméně časově náročné.

- **Stativ**

Všechny výše uvedené části jsou umístěny na stativu, díky kterému lze celý systém posunout do ideální polohy a výšky. Emulátor a čtečka karet jsou umístěny na jedné z noh stativu v plastovém pouzdře vytištěném na 3D tiskárně vyvíjené společností Y Soft.

- **Senzor papírů**

Pro správné ověření výsledku testů je na tiskárně umístěn senzor papírů, který je schopen robotovi předat informace o počtu vytištěných papírů, zda se jedná o papír velikosti A3 nebo A4 a také jestli je tisk jednostranný nebo oboustranný. Je-li použití tohoto senzoru znemožněno, lze jeho použití při testu vypnout a spoléhat se pouze na informace dodané systémem tiskárny.

Po úspěšném sestavení robota a jeho připojení do systému bylo nutné nastavit a zkaliбrovat robotickou ruku a kameru v systému RMS.

Poloha kamery je téměř libovolná, záběr však musí obsahovat celou obrazovku. Kameru je nutné zaostřit na obrazovku tak, aby byl text čitelný, ale zároveň ji nepřeostrřit kvůli artefaktům ve snímcích způsobených LCD technologií. Dále je potřeba nastavit jas kamery do ideální polohy. Kalibrace kamery spočívá v označení obrazovky zařízení na snímku kamery. Díky tomu nemusí být záběr kamery přesně kolmý k obrazovce zařízení a nemusí snímat výhradně obrazovku, což by bylo ve většině případů velmi složité, v některých nemožné. Kalibrace kamery může být automatická nebo manuální. Pro automatickou kalibraci je potřeba rohy obrazovky označit speciálními nálepkami, podle kterých IMPROC rohy rozezná a sám vytvoří výřez záběru kamery tak, aby obsahoval pouze obrazovku zařízení.

Ideální poloha robotické ruky je taková, že je její základna ve stejné rovině jako spodní hrana displeje a vzdálená asi 5 cm od ní. Tato poloha by měla zajistit, že robotická ruka dosáhne na všechna místa obrazovky a bude moci vykonat optimální dotek. Kalibrace robotické ruky, na rozdíl od kamery, nemůže být provedena automaticky a vyžaduje, aby uživatel navedl stylus do tří rohů obrazovky, konkrétně pak levého dolního, levého horního a pravého horního. Poloha čtvrtého, tedy pravého dolního rohu, je dopočítána automaticky. Navádění probíhá pomocí klávesnice, kdy klávesy A a D otáčejí základnou robotické ruky, klávesy W a S oddalují či přibližují ruku k základně, klávesy R a F zvedají a snižují stylus a pomocí kláves Q a E je stylus natáčen. Alternativním řešením je navádění pomocí chytrého telefonu, kdy místo použití klávesnice naskenuje uživatel QR kód vygenerovaný systémem RMS a ve webovém prohlížeči naviguje robotickou ruku softwarovými tlačítky. Stylus by se měl obrazovky dotýkat kolmo, aby nedocházelo k jeho sklouznutí při doteku.

Po navedení robotické ruky do požadovaného rohu a správném natočení stylusu uživatel potvrdí správný dotek. Když jsou potvrzeny všechny tři potřebné rohy, robot provede zkoušku, při které se dotkne všech čtyř rohů, tedy včetně dopočítaného. Uživatel má po dokončení zkoušky možnost kalibraci opakovat nebo potvrdit a ukončit.

## 5.2 Autentizace

Mou praxi jsem započal implementací autentizačních testů, protože se většinou jedná o ty nejjednodušší testy, pohybující se pouze na dvou obrazovkách. Díky jednoduchosti těchto testů jsem se rychle naučil pracovat s RMS a pochopil jsem, jakým způsobem psát budoucí testy.

Terminal Pro 4 podporuje přihlášení pomocí:

- Čipové karty
- Pinu
- Uživatelského jména a hesla
- Různých kombinací těchto tří způsobů přihlášení

Celkem bylo vytvořeno 29 autentizačních testů, které jsou podrobněji popsány dále.

```
Copy on TP4 with default BC
[Tags]      two_robots      TP4
Given User exists in SafeQ      logout_timeout=10
And User has default BC assigned
And Paper sensor data are cleared
When Robot authenticates
And Robot navigates to copy screen
And Second robot performs a copy action
And Robot logs out
Then Copy job is accounted under BC      0      Default Project
```

*Ukázka kódu 1: Příklad testu v Robot Frameworku s Gherkin syntaxí*

Každý test v Robot Frameworku může obsahovat tagy, díky kterým je možné spouštět či naopak nespouštět určité testy. Syntaxe Robot Frameworku je velmi odlišná od Pythonu, který rozšiřuje. Na rozdíl od něj nejsou například používány čárky a závorky. Klíčová slova mohou obsahovat mezery, k oddělení klíčových slov a proměnných se používají dvě či více mezer. Testy jsou rozděleny do souborů s příponou „robot“. Každý soubor tvoří svazek testů, nazývaný „suite“ a může obsahovat několik částí, ovlivňující chování testů v něm (viz ukázka kódu 2).

- **Settings**

Definuje nastavení souboru testů a importuje případné další soubory. Mnou nejčastěji používané byly následující položky:

- **Suite setup**

Definuje klíčové slovo, které se vykoná před započítím vykonávání testů v souboru testů. V našem případě se v suite setup vykonalo přidání terminálu do systému. Díky

tomu nemusel být terminál přidáván a odebírán při každém testu, ale pouze při změně testované části. Klíčovému slovu byly v každém souboru předávány argumenty tak, aby byl terminál nainstalován s požadovaným nastavením.

- **Suite teardown**

Stejně jako u suite setupu definuje klíčové slovo vykonané pouze jednou za spuštění všech testů v souboru, na rozdíl od něj se však jedná o klíčové slovo spuštěné po vykonání všech testů. Ve všech vytvořených testech se jednalo o klíčové slovo, které odstranilo případné změny provedené v systému a odinstalovalo terminál ze systému.

- **Test setup**

Definuje výchozí klíčové slovo, vykonané před každým testem. Tento parametr platí pouze pro testy, u kterých není definovaný jiný test setup. V našem případě se v test setup vždy provedla kontrola připravenosti robotického systému a vytvořilo se spojení s RMS, pomocí kterého byly robotické ruce předávány pokyny. Případné další funkce definované v test setup záležely na zaměření souboru testů.

- **Test teardown**

Definuje výchozí klíčové slovo, vykonané po dokončení každého testu. Stejně jako Test setup platí pouze pro testy, u kterých není definován jiný test teardown. Toto klíčové slovo bylo používáno hlavně pro odstranění vytvořeného uživatele ze systému a správné ukončení testu v RMS. Pokud by se test neukončil, nebylo by možné spustit další testy.

- **Force tags**

Definuje tagy aplikované na všechny testy v souboru testů.

- **Library**

Pomocí tohoto parametru je možné importovat python soubor a používat tak funkce v něm definované.

- **Resource**

Pomocí tohoto parametru je možné importovat robot framework soubor a používat klíčová slova a proměnné v něm definovaná.

- **Test template**

Tento parametr je použitý, je-li celý obsah souboru testů opakující se test, pouze s pozměněnými parametry. Definované klíčové slovo je pak spouštěno pro každý test se zadanými parametry a není třeba jej u každého testu znova psát.

- **Variables**

V této části souboru jsou definovány proměnné, aplikovatelné na všechny testy v souboru testů. Proměnné jsou dosažitelné i v klíčových slovech, spuštěných ze souboru testů. Proměnné v Robot Frameworku jsou označeny znakem dolaru následovaným názvem proměnné ve složených závorkách, tedy například „\${VARIABLE}“. Celá syntaxe Robot Frameworku je „case insensitive“, nezáleží tedy na tom, zda používáme velká či malá písmena.

- **Test cases**

V této části jsou napsány samotné testy. Při mé praxi jsem pro psaní testů používal Gherkin syntaxi, která usnadňuje a zlepšuje čitelnost testů pro uživatele, ale nemá žádný vliv na funkčnost testu (viz ukázka kódu 1). Princip Gherkin syntaxe je následující:

- Test je rozdělen na tři části:
  - prekvizity neboli předpoklady před započítím testu jsou označeny slovem „Given“;
  - samotné kroky testu jsou označeny slovem „When“;
  - závěr testu, tedy skutečnosti, očekávané po dokončení testu, jsou označeny slovem „Then“.
- Je-li v některé z částí více položek, jsou odděleny slovy „And“ nebo „But“, jedná-li se o zápor.

Testy mohou obsahovat již zmíněné tagy, pomocí kterých je možné spustit pouze požadované testy nebo naopak nespouštět testy s určitým tagem. V případě, že je průběh testu stejný pro více testů, je možné použít template neboli vzor. Pomocí těchto vzorů lze jednoduše spouštět stejné klíčové slovo s jinými argumenty bez nutnosti tento test vypisovat vícekrát.

- **Keywords**

Poslední z částí souboru obsahuje klíčová slova, která jsou obdobou funkcí ve většině programovacích jazyků. Cílem rozdělení testů a klíčových slov je udržení testů syntakticky jednoduchých a uživatelsky čitelných i pro uživatele, kteří nejsou s Robot Frameworkem a programováním celkově seznámeni.

Spouštění testů v Robot Frameworku může být ovlivněno pomocí následujících příkazů v příkazovém řádku:

- **-test/-t**

Spustí pouze testy, jejichž název odpovídá zadanému textu.

- **-suite/-s**

Spustí pouze testy v souborech testů, jejichž název odpovídá zadanému textu.

- **-include/-i a -exclude/-e**

Spustí pouze testy, jejichž tagy obsahují či naopak neobsahují určité tagy. Include a exclude je možné kombinovat pro přesnější určení spouštěných testů.

V argumentech je možné používat logické operátory a regex výrazy. Spuštění testů s parametry „-include xORy -exclude z“ tedy spustí testy, které obsahují tag „x“ nebo „y“ a zároveň neobsahují tag „z“.

```
*** Settings ***
Suite Setup      Suite Setup - Regression tests      authentication method=CARD
Suite Teardown    Suite Teardown - Regression tests
Test Setup        Test Setup - Tests with invalid card
Test Teardown     Test Teardown - Tests with robot
Force Tags        FX_XCP      Ricoh      Ricoh_SOP      Xerox_1stGen      TP4
Resource          ../../resources/Robotic_lab_tests/authentication_templates.robot

*** Variables ***
${PAYMENT}        ${False}

*** Test Cases ***
Authentication with Card
    [Tags]          safeQube      Brother      KM_1stGen      KM_2ndGen      KM_IPM
    [Template]       Authentication with 1 method
    Card

Logout with card
    Given User exists in SafeQ
    When Robot authenticates with card
    And Robot logs out with card
    When Robot authenticates with card
    And Robot logs out with unregistered card
    Then Terminal access log exists

*** Keywords ***
Robot logs out with card
    _Run semiautomated keyword      'Swipe card and verify Login Screen'
    ...      Emulate card
    _Ignore fail if semiautomated      Verify Screen or Throw      Login Screen
    ...      Logout with card failed, Login screen was not verified
```

*Ukázka kódu 2: Příklad \*.robot souboru*

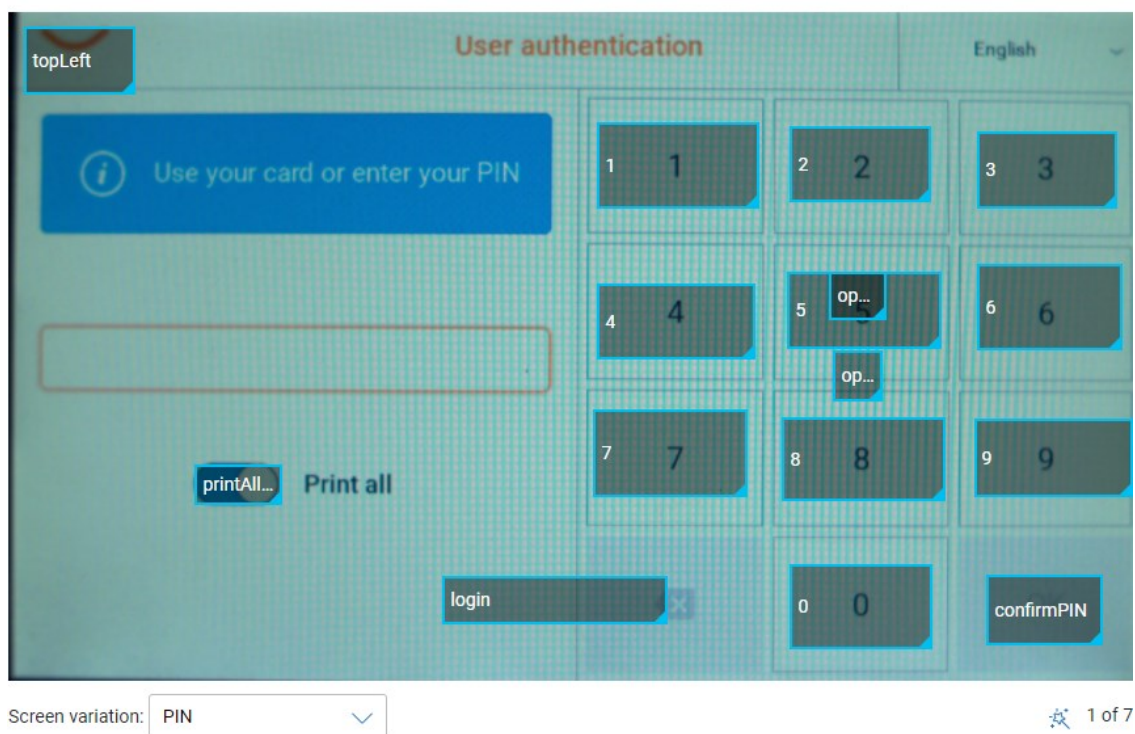
Autentizační testy vyžadují různé varianty přihlašovací obrazovky a jakoukoliv obrazovku s příznakem „vyžadována autentizace“, v případě přihlášení pomocí jména a hesla je zapotřebí třetí obrazovka obsahující softwarovou alfanumerickou klávesnici, pomocí které robot napíše přihlašovací jméno uživatele a jeho heslo.

Bylo tedy nutné tyto obrazovky zavést do systému RMS. Přidávání obrazovek je velmi jednoduché, ve webovém rozhraní RMS stačí vybrat ze seznamu požadovanou kameru a kliknutím na tlačítko vytvořit snímek. Po pojmenování a popsání snímku je automaticky spuštěno trénování IMPROC, aby byl schopen novou obrazovku rozeznat. V případě potřeby může uživatel upravit vlastnosti obrazovky, jako například chování při setkání s obrazovkou nebo zda je potřeba autentizace.

## 5.2.1 Jednofaktorová

### 5.2.1.1 PIN

Začal jsem implementací nejjednoduššího způsobu přihlášení, což je přihlášení pomocí pinu. Princip přihlášení pomocí pinu spočívá v tom, že je každému uživateli přiděleno 4 a vícemístné unikátní číslo. Shoduje-li se zadané číslo s číslem přiřazeným některému z uživatelů, systém tohoto uživatele přihlásí. K testování tohoto způsobu přihlášení je potřeba, aby na obrazovce „login screen“ byly elementy pojmenované 0 až 9, pomocí kterých je robot schopen napsat jakýkoliv číselný pin (viz obrázek 2).



Obrázek 2: Elementy na obrazovce „Login screen“, varianta „PIN“

Soubor testů ověřující přihlášení pomocí pinu obsahuje pouze dva testy:


- **Authentication with PIN**


Spustí flow, jejíž součástí je zadání pinu a jeho potvrzení. Po dokončení následuje ověření cílové obrazovky, tedy jakékoliv obrazovky s nastaveným příznakem „vyžadována autentizace“. Po úspěšném ověření se robot z terminálu odhlásí a zkontroluje, zda existuje v systému záznam o přihlášení pro daného uživatele. Pin je napsán a potvrzen pomocí flow.


- **Authentication with incorrect and correct PIN**

Jedná se o obdobu předchozího testu, na rozdíl od něj je však před zadáním správného pinu zadán náhodně vygenerovaný pin a je ověřeno, zda došlo k odmítnutí přihlášení a terminál tedy zůstal na přihlašovací obrazovce.

**Flow View**


 enterPIN


 Fictive



☐ Verify destination screen

Source screen: Login Screen



 confirmPIN

☐ Verify destination screen

Destination screen: SQ Print

**Flow with ID: 558**

Flow type

AuthenticationPin

Domains

SQTA

SQTASQ6T

RELOAD FLOW

ADD STEP

SAVE FLOW

Obrázek 3: Flow autentizace pomocí pinu pro Terminal Pro 4

#### 5.2.1.2 Čipová karta

Dalším testovaným způsobem přihlášení bylo přihlášení pomocí čipové karty. Tento způsob přihlášení vyžaduje funkční čtečku a emulátor čipových karet. Přihlašovací obrazovka při tomto způsobu přihlášení zobrazuje animaci, naznačující přiložení čipové karty ke čtečce karet. Pro správné rozpoznání obrazovky bylo tedy třeba přidat do RMS několik snímků obrazovky zachycující animaci v různých pozicích.

Soubor testů použitý k testování tohoto způsobu přihlášení obsahuje následující tři testy:

- **Authentication with card**

Při tomto testu je simulováno přiložení čipové karty a ověřeno, zda bylo přihlášení úspěšné. Po ověření se robot z terminálu odhlásí a je zkontrolováno, zda existuje v systému záznam o přihlášení uživatele ze zařízení.

- **Authentication with incorrect and correct card**

Při tomto testu je vygenerováno náhodné číslo karty, jehož přiložení je následně simulováno emulátorem čipových karet. Poté, co robot ověří, že se přihlášení nezdařilo, je emulovaná karta přiřazená uživateli a je ověřeno, že se přihlášení zdařilo a existuje o něm v systému záznam.

- **Logout with card**

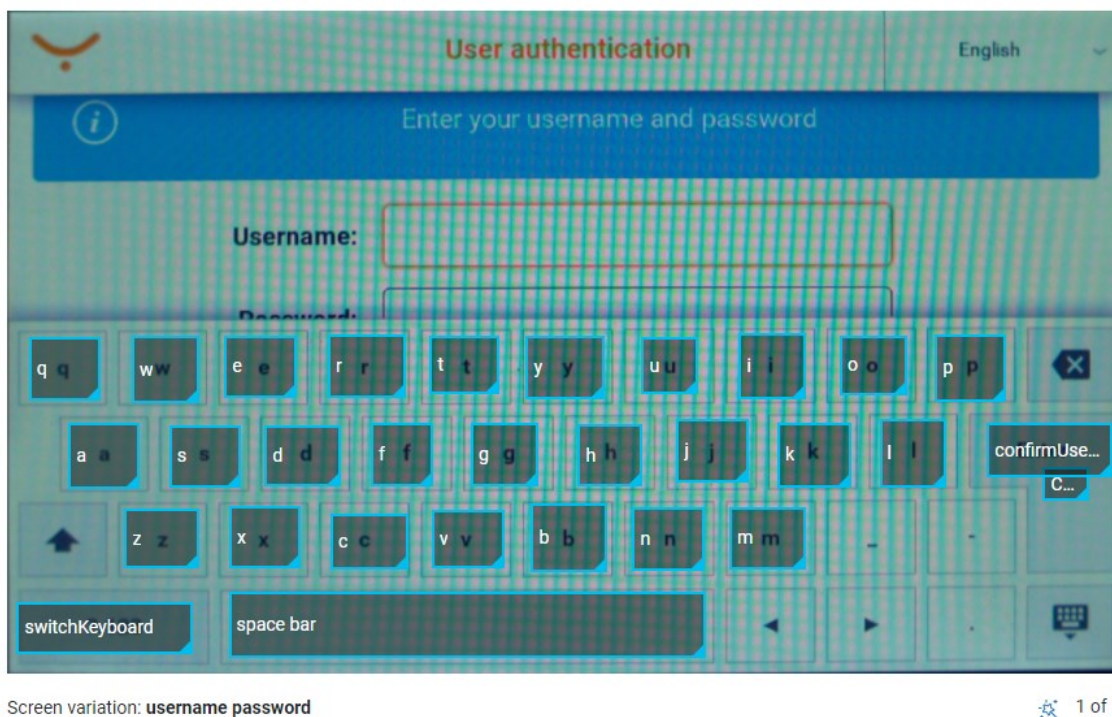
Tento test je jediným testem, který nevyužívá robotickou ruku. Jeho úkolem je ověřit funkci systému umožňující odhlášení uživatele pomocí přiložení čipové karty. Robot se simulováním přiložení karty přihlásí a ověří, zda bylo přihlášení úspěšné. Po ověření znovu simuluje přiložení stejné karty a ověří, zda je ze systému odhlášen



a nachází se na přihlašovací obrazovce. Poté je celý postup opakován, tentokrát však pro odhlášení použije kartu s náhodně vygenerovaným číslem.

### 5.2.1.3 Uživatelské jméno a heslo

Posledním jednoduchým způsobem přihlášení podporovaným Terminal Pro 4 je přihlášení pomocí uživatelského jména a alfanumerického hesla. Testování tohoto způsobu přihlášení je mnohem složitější než u předchozích dvou způsobů, protože na rozdíl od nich je navíc používána obrazovka klávesnice, pro kterou je zapotřebí vytvořit elementy pro každý znak na alfanumerické klávesnici (viz obrázek 4).



Obrázek 4: Elementy na obrazovce klávesnice

Pro zadání uživatelského jména a hesla je opět použita flow (viz obrázek 5), která zajišťuje kompatibilitu testu s vícero zařízeními, přestože je u každého z nich postup přihlášení jiný.

Soubor testů, ověřujících přihlášení pomocí jména a hesla, obsahuje pouze dva testy:

- **Authentication with username password**  
Při tomto testu je spuštěna již zmíněná flow (viz obrázek 5). Průběh testu je následující:
  - robot klikne do textového pole pro zadání uživatelského jména, což vyvolá zobrazení klávesnice;
  - robot zadá uživatelské jméno a stiskne klávesu enter, čímž uživatelské jméno potvrdí a dostane se tak zpět na obrazovku „login screen“;

- robot akci opakuje, tentokrát však pro textové pole pro zadání hesla;
  - zadané údaje jsou potvrzeny;
  - robot ověří, zda se nachází na autentizované obrazovce;
  - robot se odhlásí a ověří, zda existuje záznam o přihlášení do terminálu pro daného uživatele.
- **Authentication with incorrect and correct username password**  
Tento test je opět obdobou předchozího testu, před zadáním správných přihlašovacích údajů a ověřením přihlášení je však zadáno náhodně vygenerované uživatelské jméno a heslo a je ověřeno, zda přihlášení selhalo.

## 5.2.2 Dvoufaktorová

Terminal Pro 4 umožňuje nastavit způsob přihlášení tak, že je k přihlášení zapotřebí čipová karta a pin nebo čipová karta a uživatelské jméno a heslo. Díky tomuto dvoufaktorovému způsobu přihlášení je zvýšena bezpečnost terminálu za cenu pohodlnosti.

### 5.2.2.1 Čipová karta a pin

Výchozí obrazovka na Terminal Pro 4 je při tomto způsobu přihlášení stejná, jako při přihlašování kartou, po přiložení čipové karty je obrazovka totožná jako při přihlašování pinem. Nebylo tedy nutné přidávat nové obrazovky.

Při implementaci těchto testů jsem narazil na první problém se stabilitou způsobený nestabilní emulací čipové karty. Robot nerozlišuje mezi jednotlivými variantami obrazovek, což znamená, že je pro něj varianta obrazovky „login screen“ zobrazená při přihlášení kartou a varianta při přihlášení pinem tatáž obrazovka. Po přiložení karty a změně obrazovky na variantu zobrazující přihlášení pinem tak nevidí mezi obrazovkami rozdíl a správně předpokládá, že nedošlo ke změně obrazovky. Tato skutečnost zamezuje odhalení chyby při emulaci čipové karty a robot se tak nesnaží chybu napravit opakovanou emulací. Stabilita těchto testů tak byla zdokonalena odstraněním příčiny, tedy umístěním čtečky a emulátoru tak, aby bylo dosaženo stoprocentní spolehlivosti.

Soubor testů ověřující tento způsob přihlášení obsahuje 4 testy:

- **Authentication with correct Card and PIN**  
Při tomto testu provede robot přihlášení pomocí správné čipové karty a pinu, přiřazených uživateli. Přihlášení je ověřeno a je zkontrolován záznam o přihlášení.
- **Authentication with incorrect Card and correct PIN**  
U tohoto testu je simulováno přiložení čipové karty s náhodně vygenerovaným číslem. Vzhledem k tomu, že je karta i pin ověřena souběžně až po potvrzení pinu, se terminál

chová při přiložení špatné čipové karty stejně jako při přiložení správné a vyzve tedy uživatele k zadání pinu. Robot pin zadá a ověří, zda přihlášení selhalo.

- **Authentication with correct Card and incorrect PIN**

Průběh tohoto testu je stejný jako u předchozího, tentokrát je však simulováno přiložení správné karty a zadán náhodně vygenerovaný pin.

- **Authentication with incorrect Card and incorrect PIN**

Tento test má stejný průběh jako předchozí dva zmíněné testy, číslo karty i pin jsou však náhodně vygenerované.

## Flow View

Source screen: **Login Screen**

openUsername

☒ Verify destination screen

Destination screen: **Keyboard**

enterUsername **Fictive**

☐ Verify destination screen

Source screen: **Keyboard**

confirmUsername

☒ Verify destination screen

Destination screen: **Login Screen**

Source screen: **Login Screen**

openPassword

☒ Verify destination screen

Destination screen: **Keyboard**

enterPassword **Fictive**

☐ Verify destination screen

Source screen: **Keyboard**

confirmUsername

☒ Verify destination screen

Destination screen: **Login Screen**

Source screen: **PIN**

login

☐ Verify destination screen

Destination screen: **SQ Print**

## Flow with ID: 559

Flow type

AuthenticationUserPass

Domains

SQTA

SQTASQ6T

RELOAD FLOW

ADD STEP

SAVE FLOW

Obrázek 5: Flow přihlášení pomocí uživatelského jména a hesla

#### 5.2.2.2 Čipová karta a uživatelské jméno a heslo

Bezpochyby nejbezpečnější způsob přihlášení, ale zároveň nejtěžší na implementaci testů. Princip tohoto způsobu přihlášení je stejný jako u autentizace pomocí čipové karty a pinu, rozdílná je pouze obrazovka, tedy i akce po přiložení čipové karty.

Soubor testů ověřující tento způsob přihlášení má také 4 testy, jejichž princip je shodný s předchozím.

### 5.2.3 Více možností autentizace

Další kombinace základních tří způsobů přihlášení, která na rozdíl od dvoufaktorového přihlášení zvyšuje uživatelské pohodlí za cenu potencionálně snížené bezpečnosti. Princip těchto způsobů přihlášení je takový, že má uživatel na výběr mezi dvěma možnostmi a k přihlášení mu stačí jedna z nich.

#### 5.2.3.1 Čipová karta nebo pin

Výchozí obrazovka při tomto způsobu přihlášení je totožná s obrazovkou při přihlašování pinem, liší se pouze v textu, informujícím uživatele o možnosti přihlášení kartou. Tento rozdíl v textu však není dostatečně výrazný na to, aby byla obrazovka považována robotem za jinou obrazovku.

Soubor testů ověřující tento způsob přihlášení obsahuje 5 testů:

- **Authentication with correct Card (Card or PIN)**  
Robot simuluje přiložení správné karty a ověří přihlášení a záznam o něm.
- **Authentication with correct PIN**  
Robot se přihlásí pomocí správného pinu, ověří přihlášení a záznam o něm.
- **Authentication with incorrect Card or correct PIN**  
Robot simuluje přiložení neznámé karty a ověří neúspěch, poté se přihlásí pomocí správného pinu, ověří přihlášení a záznam o něm.
- **Authentication with correct Card or incorrect PIN**  
Robot se pokusí o přihlášení pomocí nesprávného pinu a ověří neúspěch, poté se přihlásí pomocí správné čipové karty, ověří přihlášení a záznam o něm.
- **Authentication with incorrect Card or incorrect PIN**  
Robot se pokusí o přihlášení pomocí nesprávného pinu a ověří neúspěch, poté se pokusí o přihlášení pomocí nesprávné čipové karty a opět ověří neúspěch.

#### 5.2.3.2 Čipová karta nebo uživatelské jméno a heslo

Princip tohoto způsobu přihlášení je téměř totožný s přihlášením pomocí čipové karty nebo pinu, uživatel má ale na výběr mezi přihlášením kartou a přihlášením pomocí uživatelského jména a hesla.

Soubor testů ověřujících tento způsob přihlášení obsahuje taktéž 5 testů, jejichž princip je totožný s testy ověřujícími přihlášení pomocí čipové karty nebo pinu.

#### 5.2.4 Přiřazení čipové karty

Terminal Pro 4 umožňuje uživatelům, aby si sami přiřadili čipovou kartu. Tato funkce musí být povolena v systémovém nastavení SAFEQ a je ověřována posledním souborem testů z autentizační části.

Přiřazení čipové karty může probíhat pomocí uživatelského jména a hesla nebo aktivačního kódu karty, zaslaného uživateli na email.

Soubor testů ověřujících přiřazení čipové karty obsahuje 4 testy:

- **Card assignment with Login and Password**

Test ověřující přiřazení čipové karty pomocí uživatelského jména a hesla. Robot simuluje přiložení neznámé čipové karty, zadá uživatelské jméno a heslo a pokusí se s aktivovanou kartou přihlásit.

- **Card assignment with Card activation code**

Robot zažádá o přiřazení aktivačního kódu karty, který je mu následně zaslán na email, ze kterého je vyčten. Poté simuluje přiložení neznámé čipové karty, kterou aktivuje získaným kódem a ověří, zda se může nově aktivovanou kartou přihlásit.

- **Card assignment with Card activation code – One card already assigned**

Jedná se o obdobu předchozího testu, tentokrát však se systémovým nastavením, které nedovoluje uživatelům mít přiřazeno více než jednu čipovou kartu. Je tedy vytvořen uživatel s již přiřazenou kartou, poté je aktivována nová karta a ověřeno, zda má uživatel přiřazenou pouze jednu kartu, kterou je ta aktivovaná robotem.

- **Usage of Card activation code for several cards**

Tento test ověřuje skutečnost, že je každý aktivační kód možné použít pouze jednou. Jedná se o dvakrát po sobě jdoucí testy shodné s druhým vyjmenovaným testem, z nichž první musí být úspěšný a druhý neúspěšný.

## 5.3 Tisk

Bezpochyby nejdůležitější funkcí SAFEQ je tisk dokumentů a jejich zaúčtování. Terminal Pro 4 podporuje 3 způsoby účtování tiskových úloh:

- **Offline účtování**

Monitoruje všechny tiskové úlohy zaslané ze serveru SAFEQ do tiskárny, dokud nejsou vytištěny. Přesnost tohoto způsobu účtování není vysoká, protože nebere v potaz stav zařízení a výstup tiskárny. Vytisknutí barevného dokumentu na tiskárně podporující pouze černobílý tisk tak při použití offline účtování povede k zaúčtování barevného tisku, přestože uživatel obdržel černobílý tisk.

- **Online účtování**

Jedná se o přesnější formu monitorování aktivity tiskárny. Hlavní rozdíl je ve vyčkání na zaúčtování úlohy po dokončení tisku. Tato metoda účtování vyžaduje nízkou odezvu mezi tiskárnou a serverem SAFEQ a použití ovladače účtování, specifického pro danou tiskárnu.

- **Žádné účtování**

Při zvolení této možnosti nejsou tiskové úlohy sledovány a zúčtovávány.

Dále podporuje dva způsoby tisku:

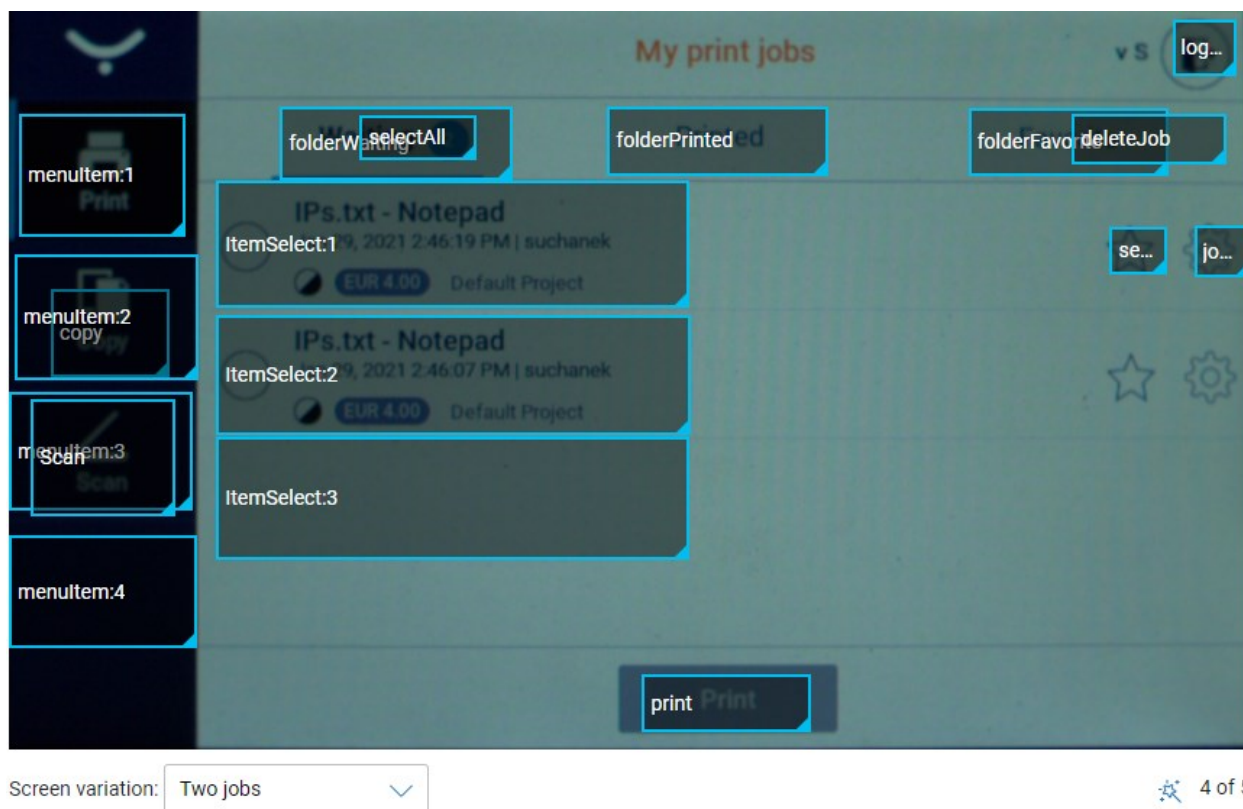
- **Autentizovaný tisk**

Pro vytisknutí dokumentu se uživatel musí na terminálu přihlásit, což zamezuje nepovolenému nahlédnutí do vytisknutých dokumentů nebo například neúmyslnému ponechání dokumentů v tiskárně.

- **Přímý tisk**

Dokumenty jsou vytištěny ihned po odeslání do tiskové fronty bez nutnosti přihlášení, ale stále jsou zaúčtovány a zaevidovány v systému SAFEQ.

Bylo tedy nutné přidat do RMS další obrazovku pojmenovanou „SQ Print“, pomocí které lze na terminálu tisknout dokumenty, a pomocí ní otestovat všechny tři podporované způsoby účtování v kombinaci s oběma způsoby tisku.



Obrázek 6: Elementy na obrazovce tisku bez zvolené tiskové úlohy

Celkem bylo vytvořeno či přizpůsobeno 25 testů ověřujících funkci tisku, mezi nejvýznamnější patří:

- **Secure print with accounting**

Tento test využívající vzor se objevuje ve třech variantách, tedy pro všechny tři způsoby účtování. Jedná se o základní test ověřující funkčnost autentizovaného tisku, při kterém je vytvořen nový uživatel, kterému je do tiskové fronty přidána tisková úloha. Robot po autentizaci zvolí první tiskovou úlohu a její zvolení je pro zvýšení stability testu ověřeno. Poté je tisková úloha vytištěna, její úspěšné vytištění je ověřeno pomocí senzoru papírů. Je také ověřeno její správné zaúčtování v SAFEQ.

- **Direct print with accounting**

Tento test se objevuje pouze ve dvou variantách, a to pouze pro offline účtování a žádné účtování, protože online účtování není s přímým tiskem kompatibilní. Vzhledem k povaze přímého tisku není při tomto testu používána robotická ruka. Při přímém tisku totiž není vyžadována interakce uživatele s terminálem. Je tedy pouze zaslána tisková úloha vytvořenému uživateli do fronty přímého tisku. Úspěšné vytištění dokumentu je opět ověřeno pomocí senzoru papírů a je také ověřeno správné zaúčtování tiskové úlohy v SAFEQ.



- **Secure print with accounting using print all**

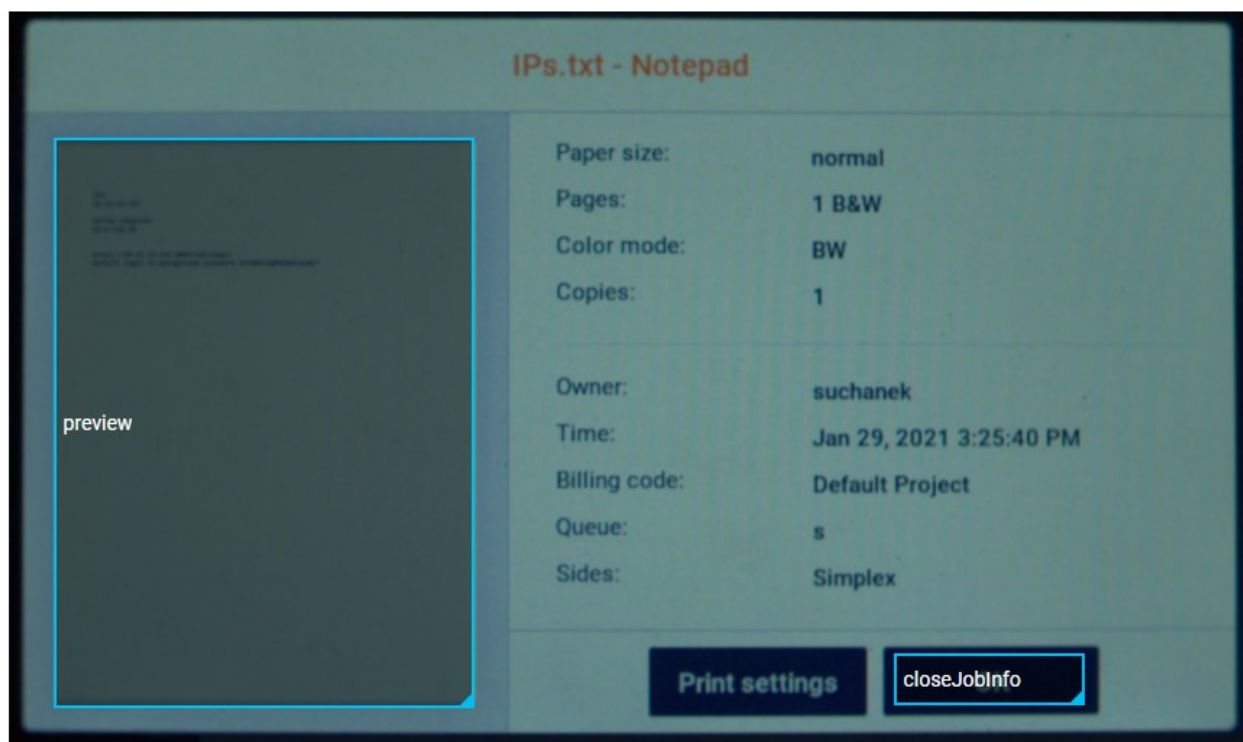
Jedná se o test ověřující funkci tlačítka „print all“ na autentizační obrazovce viditelného na obrázku 2. Je-li toto tlačítko aktivováno před autentizací uživatele, jsou po úspěšném přihlášení automaticky vytištěny všechny nevytištěné tiskové úlohy. Nově vytvořenému uživateli jsou zaslány do fronty bezpečného tisku úlohy, robot aktivuje tlačítko „print all“ a přihlásí se. Správná funkce je opět ověřena pomocí senzoru papírů a je ověřeno správné zaúčtování tiskové úlohy v SAFEQ.

- **Secure print from favorite folder with accounting**

SAFEQ umožňuje uživatelům označit tiskové úlohy jako oblíbené. Takto označené úlohy jsou označený příznakem oblíbené úlohy, jsou dostupné ze složky oblíbených a nejsou smazány po vytištění. Uživatel tak nemusí pokaždé úlohu do tiskové fronty posílat, jedná-li se například o často tištěný dokument. Při tomto testu je vytvořen nový uživatel, kterému je do tiskové fronty zaslána tisková úloha. Po přihlášení ji robot označí tlačítkem znázorňujícím hvězdu viditelném na obrázku 6 jako oblíbenou, přesune se do složky oblíbených, kde by měl být dokument viditelný, a poté jej vytiskne.

- **Mark waiting job as favorite and display preview**

Tento test slouží k ověření funkčnosti oblíbených úloh a správného zobrazení náhledu úlohy. Přiřazená tisková úloha je označena jako oblíbená a robot se přesune do složky oblíbených tiskových úloh. Kliknutím na tlačítko znázorňující ozubené kolo (viz obrázek 6) se robot dostane na obrazovku zobrazující detail tiskové úlohy (viz obrázek 7). Vzhledem k tomu, že je náhled příliš malý na to, aby robot rozeznal text, který zobrazuje, je náhled souboru předem pozměněn na obrázek obsahující pouze modrou barvu. Pomocí IMPROC se pak ověří, zda barva v elementu nazvaném „preview“ je modrá. Při tomto testu nedochází k tisku dokumentů, pouze je ověřeno v SAFEQ, zda je daná tisková úloha označena jako oblíbená.



Screen variation: **Job Info**

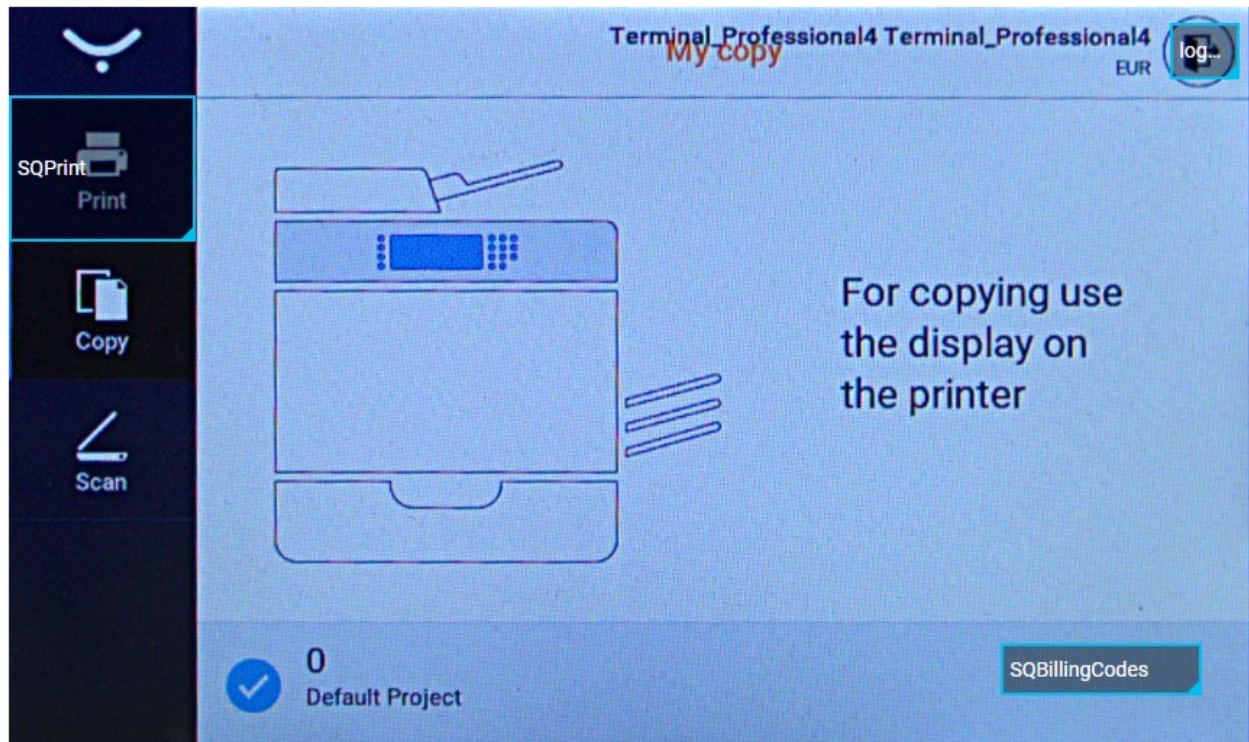
1 of 1

Obrázek 7: Elementy na obrazovce detail úlohy

Jak již bylo řečeno, má práce se od mých předchůdců lišila zejména v tom, že testování neprobíhalo na zařízení zabudovaném v tiskárně, ale na externím zařízení, které je schopno s tiskárnou pouze omezeně komunikovat. Chce-li uživatel pomocí Terminal Pro 4 kopírovat či skenovat dokumenty, je odkázán na terminál tiskárny (viz obrázek 8). Bylo tedy nutné sestavit druhého robota umístěného u terminálu tiskárny a implementovat možnost spolupráce mezi těmito dvěma roboty. Díky tomu bylo možné plně nahradit lidskou obsluhu a automatizovat testování.

Rozhraní terminálu na připojené tiskárně zůstává původní, bylo tak nutné do RMS přidat všechny potřebné obrazovky a jejich elementy. Kvůli velmi rozdílnému průběhu skenování či kopírování v porovnání s aplikací SAFEQ na zabudovaných terminálech tiskáren bylo nutné vytvořit zcela nové testy zohledňující tuto skutečnost.

## 5.4 Kopírování



Screen variation: Copy

1 of 2

Obrázek 8: Obrazovka na Terminal Pro 4 při kopírování

Pro ověření kopírování byl vytvořen testovací vzor využívající spolupráce dvou robotů (viz ukázka kódu 3), umožňující otestovat kopírování pomocí Terminal Pro 4 s jakoukoliv variací parametrů:

- černobílý nebo barevný tisk;
- velikost papíru A3 nebo A4;
- jednostranný či oboustranný tisk;
- libovolný počet kopií.

Při spuštění tohoto testovacího vzoru provede robot umístěný u Terminal Pro 4 autentizaci a automaticky se naviguje do záložky kopírování, čímž je odblokována připojená tiskárna. Následně je jako aktivní robot zvolen robot u připojené tiskárny, který pomocí flows provede potřebná nastavení kopírování a poté jej spustí. Po dokončení kopírování je opět zvolen robot u Terminal Pro 4, který provede odhlášení z terminálu. Nakonec je zkontrolováno správné zaúčtování v SAFEQ.

```
Copy A3 BW on TP4 and check accounting
[Tags]      two_robots      TP4
[Template]   Copy on TP4 with certain configuration and check accounting
              BW      A3      2.6      1      simplex      1      A3Simplex
```

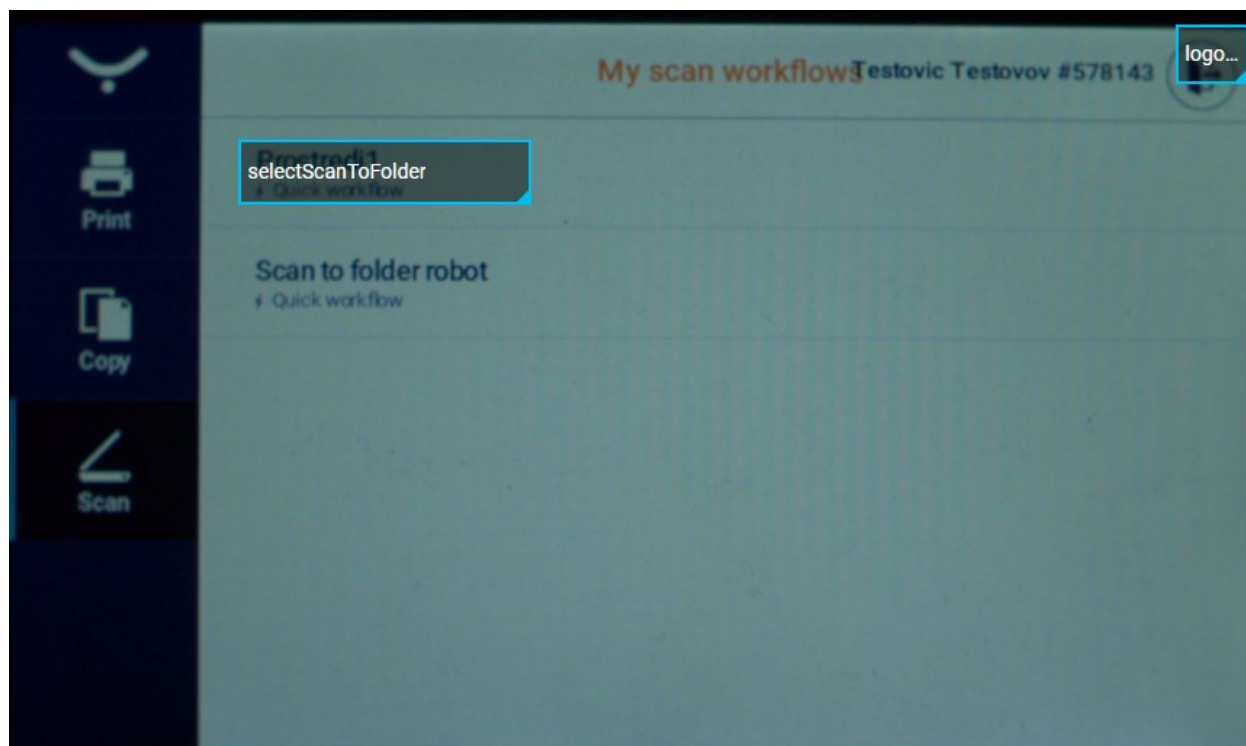
Ukázka kódu 3: Test kopírování implementovaný pomocí vzoru

## 5.5 Skenování

Poslední funkcí Terminal Pro 4, pro kterou bylo potřeba implementovat testy, bylo skenování. Skenování v SAFEQ probíhá pomocí tzv. „scan workflows“, což jsou jednoduše implementované vzory, obsahující informace zejména o požadovaném cíli skenu. Mezi nejčastěji používané patří nahrání do síťové složky a odeslání v příloze emailu uživateli.

Pro otestování skenování byl vytvořen pouze jeden test, ověřující vzor skenování do složky.

Po autentizaci na Terminal Pro 4 robot zvolí skenování a uživatel je vyzván k výběru některého z dostupných skenovacích vzorů. Vzhledem ke konstantnímu názvu vzoru vytvořeného robotem jsem zvolil přístup dynamického umístění elementu, což zaručuje, že robot pokaždé klikne na správný vzor v nabídce, tedy vzor nazvaný „Scan to folder robot“, přestože se jeho poloha v seznamu může měnit.

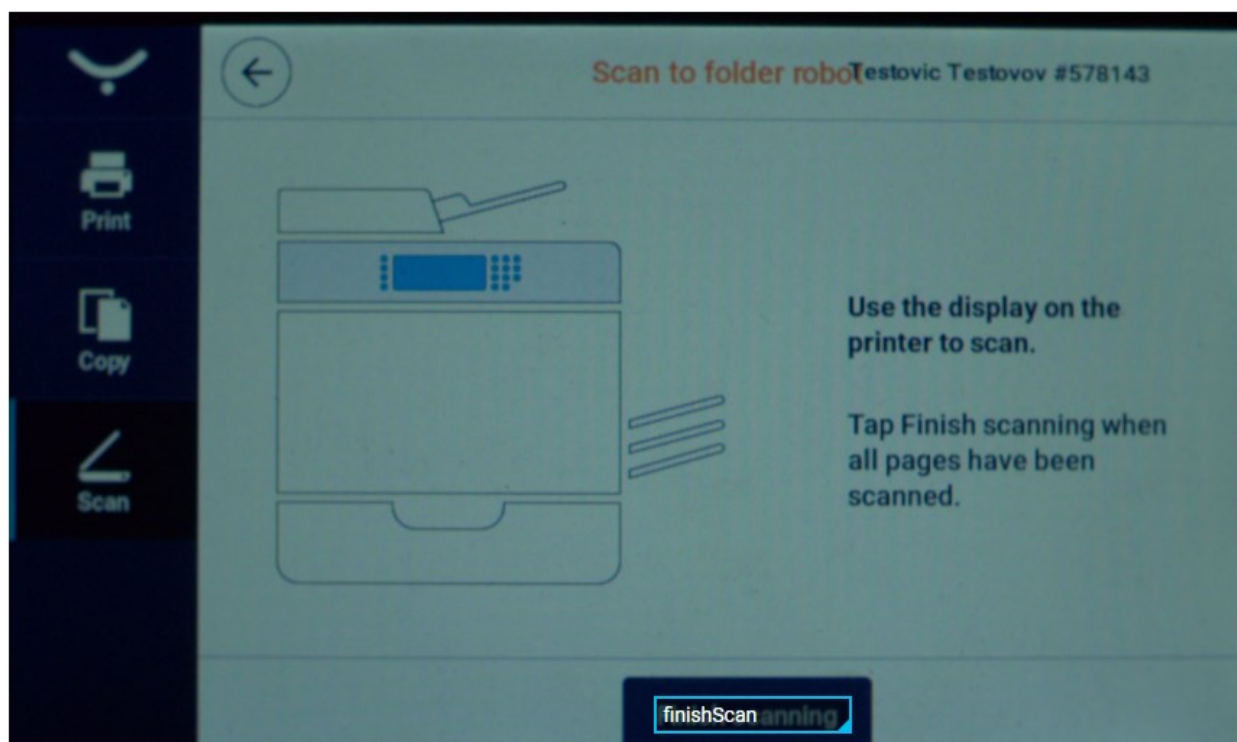


Screen variation: two workflows

1 of 1

Obrázek 9: Obrazovka volby skenovacího vzoru

Po zvolení vzoru je uživatel vyzván, aby provedl skenování na zabudovaném terminálu tiskárny a poté stisknul tlačítko, které dokončí sken (viz obrázek 10).



Screen variation: Scan in progress

1 of 1

Obrázek 10: Obrazovka probíhajícího skenování

Po ověření této obrazovky je aktivován druhý robot, umístěný u připojené tiskárny, který za pomoci flow provede skenování na terminálu tiskárny. Poté robot u Terminal Pro 4 potvrdí dokončení skenování a je ověřena existence souboru v nastavené složce a zaúčtování skenu v SAFEQ.

## 5.6 Nastavení terminálu

Nastavení terminálu je možné pouze pomocí servisního menu, ke kterému získá uživatel přístup po sekvenci stisknutí skrytých tlačítek a zadání servisního pinu. Vzhledem k téměř totožnému vzhledu přihlašovací obrazovky ve variantě pin a obrazovky přihlášení do servisního menu bylo zapotřebí použít textový distinguisher, který podle informačního nápisu na obrazovce určil, o kterou z nich se jedná.

Bylo vytvořeno 5 testů ověřujících nastavení terminálu:

- **Service menu entry and logout**

Test ověřující fungování skrytých tlačítek a správného přístupu do servisního menu. Po úspěšné autentizaci a ověření obrazovky se robot ze servisního menu odhlásí a ověří jej.

- **Changes of DHCP settings**

Tento test ověřuje správné fungování síťového nastavení, konkrétně nastavení DHCP. Robot se stisknutím skrytých tlačítek a zadáním servisního pinu dostane do servisního menu, kde v záložce síťového nastavení vypne DHCP. Terminal Pro 4 je dálkově restartován a pomocí webového rozhraní je síťové nastavení ověřeno. Poté je postup opakován, tentokrát je však DHCP robotem zapnuto.

- **Changes of service menu PIN**

Pomocí tohoto testu je ověřena správná funkce možnosti změny servisního pinu. Robot se autentizuje do servisního menu a zvolí ze seznamu možnost změny servisního pinu. Změna je ověřena zadáním současného servisního pinu. Poté je potřeba zadat nový pin a správné zadání ověřit opětovným zadáním nového pinu. Robot se ze servisního menu odhlásí a opět se přihlásí, tentokrát za použití nového servisního pinu. Úspěch autentizace je ověřen a robot změní servisní pin zpět na původní.

- **Available languages**

Tento test neprobíhá v servisním pinu, ale na přihlašovací obrazovce, na které robot klikne na element volby jazyka a pomocí rozpoznání textu zkontroluje, zda jsou k dispozici všechny podporované jazyky.

## 6. Závěr

Vykonání bakalářské práce formou odborné praxe hodnotím velice kladně. Dozvěděl jsem se díky ní, jak probíhá práce v týmu a ve větších firmách a získal tak představu o mém budoucím profesním životě. Práce s robotem pro mě byla velice zajímavá a rád bych se tomuto odvětví věnoval i v budoucnu. Nabyl jsem také velkou řadu cenných vědomostí, zejména o reálné práci na projektech, která je od semestrálních projektů velmi odlišná. Naučil jsem se také, jak psát udržitelný a čitelný kód tak, aby byl čitelný, lehce pochopitelný a v budoucnu rozšiřitelný ostatními členy týmu.

Během mé odborné praxe jsem zužitkoval spoustu znalostí získaných v rámci studia. Jednalo se především o základy programování v jazyce Python z předmětu Uživatelská rozhraní, síťovou komunikaci z předmětu Počítačové sítě, jazyk HTML, XML a JSON soubory z předmětu Vývoj internetových aplikací.

Vzhledem k velkému množství interních systémů jsem se musel práci s mnohými z nich učit až po nástupu do firmy a mé zaučení tedy i přes velikou ochotu a nápomocnost členů týmu i ostatních zaměstnanců trvalo déle. Další chybějící znalostí byla práce s verzovacím systémem Git, který je neodmyslitelnou součástí práce v týmu. Musel jsem se také naučit pracovat s Robot Frameworkem a Gherkin syntaxí.

Doufám, že má práce splní svůj účel a ušetří softwarovým inženýrům mnoho hodin únavného testování.



## 7. Literatura

- [1] Y Soft Corporation – Intelligent Enterprise Office Solutions. [online]. Copyright © 2021 Y Soft Corporation. All rights reserved. [cit. 16.03.2021]. Dostupné z: <https://www.ysoft.com/cs>
- [2] YSoft SAFEQ | Y Soft Corporation. [online]. Copyright © 2021 Y Soft Corporation. All rights reserved. [cit. 16.03.2021]. Dostupné z: <https://www.ysoft.com/cs/products/enterprise-workflow-platform/print-management/ysoft-safeq>
- [3] What is Python? Executive Summary | Python.org. *Welcome to Python.org* [online]. Copyright ©2001 [cit. 18.03.2021]. Dostupné z: <https://www.python.org/doc/essays/blurbl/>
- [4] index | TIOBE - The Software Quality Company. *Home | TIOBE - The Software Quality Company* [online]. Copyright © 2020 TIOBE Software BV [cit. 18.03.2021]. Dostupné z: <https://www.tiobe.com/tiobe-index/>
- [5] PyCharm: the Python IDE for Professional Developers by JetBrains. *JetBrains: Essential tools for software developers and teams* [online]. Dostupné z: <https://www.jetbrains.com/pycharm/>
- [6] Robot Framework. *Robot Framework* [online]. Dostupné z: <https://robotframework.org/>
- [7] Jira | Issue & Project Tracking Software | Atlassian. *Atlassian | Software Development and Collaboration Tools* [online]. Copyright © 2021 Atlassian [cit. 08.04.2021]. Dostupné z: <https://www.atlassian.com/software/jira>
- [8] Sourcetree - A free Git & Mercurial client | Atlassian. *Atlassian | Software Development and Collaboration Tools* [online]. Copyright © 2021 Atlassian [cit. 08.04.2021]. Dostupné z: <https://www.atlassian.com/software/sourcetree>
- [9] Bamboo Continuous Integration and Deployment Build Server. *Atlassian | Software Development and Collaboration Tools* [online]. Copyright © 2021 Atlassian [cit. 08.04.2021]. Dostupné z: <https://www.atlassian.com/software/bamboo>